REMARKS

Claims 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, and 37-46 are pending in this application. Claims 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, and 36 are cancelled herein without prejudice or disclaimer. Claim 46 is amended herein. Support for the amendment may be found in the claims as originally filed. No new matter has been added. Reconsideration is requested based on the foregoing amendment and the following remarks.

Response to Arguments:

The Applicants appreciate the consideration given to their arguments. The Applicants, however, are disappointed that their arguments were not found to be persuasive. The final Office Action summarizes the arguments in section 28, at page 8 as:

Applicant argues that Raz does not teach a switching method.

Here, the final Office Action has apparently focused specifically on the recitation "switching," while neglecting most of the rest of the recitation of the claims. This is submitted to be incorrect.

The Applicants also argued that neither Neches nor Raz shows "switching plural processor elements one from another," which they do not, let alone 1.) "a switching request signal detecting section for detecting a switching request signal to request switching plural processor elements one from another," 2.) "storing handover information relating to the common program which information is to be handed over from said one processor element to said another processor element," or 3.) "a stop control section for stopping the performance of said one processor element after said store control section stores said handover information into said storing section."

Furthermore, the Applicants argued that it would not have been obvious to combine Neches and Raz since Raz would have added nothing to Neches. Since all of these arguments were presented in a good faith effort to advance the prosecution, a response is requested to the other arguments as well.

The final Office Action, in particular, addressed the first of the Applicants' arguments in section 29 at page 8 by asserting:

> Raz teaches a switching method (col. 2 lines 40-60; ; col 10 line 59-col. 11 line 30) 'To fully implement the above-described load balancing technique in this system, a programmable switching mechanism is added to the data storage system. The programmable switching mechanism enables the managing host processor through commands which is send to the data storage system to

> reconfigure the data storage system by changing the set of logical volumes that are connected to each host connection.

None of the sections of Raz cited at page 8 of the final Office Action, however, shows "switching plural processor elements one from another." Raz, rather, is reconfiguring a data storage system, as noted above in the final Office Action, not "switching plural processor elements one from another." Raz does this by remapping the volumes to the host connections by adding the first volume to the first host connection. In particular, as described in Raz at column 2, lines 39-45:

> In preferred embodiments, the method also includes the steps of sending a connect command from one of the host processors to the data storage system, wherein the connect command designates a first volume and a first host connection; and in response to receiving the connect command, remapping the volumes to the host connections by adding the first volume to the first host connection.

Thus, Raz is remapping the volumes to the host connections by adding the first volume to the first host connection, not "switching plural processor elements one from another."

Raz, furthermore, remaps the volumes to the host connections by removing the first volume from the first host connection. In particular, as described in Raz at column 2, lines 45-52:

> The method also includes the steps of sending a disconnect command from one of the host processors to the data storage system, wherein the disconnect command designates a first volume and a first host connection; and in response to receiving the first disconnect command, remapping the volumes to the host connections by removing the first volume from the first host connection.

Thus, Raz is remapping the volumes to the host connections by removing the first volume from the first host connection, not "switching plural processor elements one from another."

Raz, furthermore, remaps the volumes to the host connections in response to a workload imbalance. In particular, as described in Raz at column 2, lines 52-58:

> The method further includes monitoring I/O requests that are sent to each of the volumes by each of the host processors; from monitoring information, generating workload statistics indicative of the distribution of workload among the host processors; detecting a workload imbalance; and in response thereto, remapping the volumes to the host connections.

Thus, Raz is remapping the volumes to the host connections in response to a workload imbalance, not "switching plural processor elements one from another."

Raz, finally, is using a programmable switching mechanism to reconfigure the data storage system by changing the set of logical volumes that are connected to each host connection. In particular, as described in Raz at column 10, lines 66 and 67, continuing at column 11, lines 1-5:

> To fully implement the above-described load balancing technique in this system, a programmable switching mechanism is added to the data storage system. The programmable switching mechanism enables the managing host processor through commands which it send to the data storage system to reconfigure the data storage system by changing the set of logical volumes that are connected to each host connection.

Thus, Raz is reconfiguring the data storage system by changing the set of logical volumes that are connected to each host connection, not "switching plural processor elements one from another." Even if Raz does show a switching method, Raz is still not "switching plural processor elements one from another." Raz, rather, is reconfiguring a data storage system. There are no plural processors in Raz to switch. Consequently, even if Neches and Raz were combined, as proposed in the final Office Action, the claimed invention would not result. Further reconsideration is thus requested.

Claim Rejections - 35 U.S.C. § 103:

Claims 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, and 37-46 were rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 4,445,171 to Neches, (hereinafter "Neches") in view of U.S. Patent No. 5,860,137 to Raz et al. (hereinafter "Raz"). The rejection is traversed. Reconsideration is earnestly solicited.

In several embodiments, the storing section, the store control section, the stop control section, and the start control section of independent claims 1, 44, and 45 handle handover information. The handover information is used for performing tasks carried out on two or more processors. In particular, when the control section switches a task from one processor element to another processor element, the handover information used for the performance of the task by the first processor (i.e. before switching) is also used for the performance of the task by the second processor (i.e. after switching).

Raz, in contrast, does not disclose the handover information relating to the common program recited in claims 1, 44, and 45.

Furthermore, in claims 1, 44, and 45, a single storing section is shared by two or more processor elements, so that one piece of handover information is also shared by the two or more processor elements.

In contrast to this, as shown in Fig. 2 of Raz, "HOST A" is associated with "VOL 1," "VOL 2," and "VOL 3;" , "HOST B" is associated with "VOL 4," and "VOL 5;" and "HOST C" is associated with "VOL 6," "VOL 7," and "VOL 8." That is, Raz shows no *single* volume or storing section shared among "HOST A," "HOST B," or "HOST C." Thus, even if the hosts in Raz were considered to be multiple processors, Raz still has no way to use the handover information from a first host, before switching, for the second host, after switching, when one host is switched to another host. Raz, consequently, fails to realize the following effect: the claimed invention makes it possible to easily and securely deliver the handover information of a first processor, which was used before switching, to a second processor, after switching, for performance of tasks on the second processor.

The third clause of claim 1 recites:

> Detecting a switching request signal to request switching such plural processor elements one from another.

Neches neither teaches, discloses, nor suggests switching plural processor elements one from another, let alone "detecting a switching request signal to request switching plural processor elements one from another," as acknowledged graciously in the final Office Action at page 2. The final Office Action seeks to compensate for this deficiency of Neches by combining Neches with Raz.

Raz, however, neither teaches, discloses, nor suggest switching plural processor elements one from another either, and thus cannot compensate for the deficiencies of Neches with respect to the claimed invention.

As described in Raz, rather, at column 1, lines 55-58,

> In general, in one aspect, the invention is a method of controlling distribution of processing in a system that includes a plurality of host data processors connected to a data storage system.

Thus, Raz is about distributing processing between a plurality of host data processors connected to a data storage system, not "switching plural processor elements one from another either," as recited in claim 1. All of the of host data processors in Raz will be running, Raz seeks only to equalize their ownership of volumes of a data storage system. In particular, as described in Raz, rather, at column 1, lines 58-65,

> The data storage system includes a digital memory that is partitioned into a plurality of volumes. The method includes assigning ownership of the volumes to the host processors such that each of the host processors owns a different subset of volumes. The concept of ownership means that a host processor is

> prohibited from sending I/O requests to any of the volumes which are outside of the subset of volumes which it owns.

Thus, if Raz is switching anything, he's switching the ownership of the volumes owned by each of the host processors, not "switching plural processor elements one from another" either, as recited in claim 1.

Furthermore, as described in Raz, rather, at column 1, line 65-column 2, line 6,

> The method also includes monitoring the I/O requests that are sent to each of the volumes by each of the host processors; from the monitoring information, generating workload statistics indicative of the distribution of workload among the host processors; detecting a workload imbalance; and in response to detecting the workload imbalance, reassigning ownership of the volumes to the host processors so as to change the distribution of workload among the host processors.

Thus, if Raz is switching anything, he's switching the ownership of the volumes owned by each of the host processors, not "switching plural processor elements one from another" either, as recited in claim 1. Therefore, even if Neches and Raz were combined as proposed in the final Office Action, the claimed invention would not result.

The final Office Action seeks to justify the combination of Neches and Raz at page 3 of the final Office Action by saying that,

> It would have been obvious to one skilled in the art at the time of the invention to combine the teachings of Neches and Raz because Raz's method of switching between processors would improve Neches system by allowing more than one processor to work on a task thereby improving the overall system.

Neches, however, already allows more than one process to work on a task. Neches, in particular, describes in the Abstract,

> A multiprocessor system intercouples processors with an active logic network having a plurality of priority determining nodes. Messages are applied concurrently to the network in groups from the processors and are sorted, using the data content of the messages to determine priority, to select a single or common priority message which is distributed to all the processors with a predetermined total network delay time.

Since Neches applies messages *concurrently* to a network of intercoupled multiprocessors, Neches *already* allows more than one process to work on a task. Thus, even if Raz did teach switching plural processor elements one from another either, it is submitted that persons of ordinary skill in the art would have not been motivated to combine Neches with Raz, as proposed in the final Office Action, since to do so would have added nothing to Neches.

The fifth clause of claim 1 recites:

> Storing handover information relating to the common program which information is to be handed over from said one processor element to said another processor element.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above. Since neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, they cannot show "storing handover information relating to the common program which information is to be handed over from said one processor element to said another processor element," as recited in claim 1.

Raz, rather, describes at column 13, lines 1-6,

> At said data storage system, receiving a remapping instruction from any one of said plurality of external processors; and
> In said data storage system, in response to receiving said remapping instruction Remapping said plurality of volumes to said plurality of connections.

Thus, Raz is describing remapping volumes of a data storage system, not "storing handover information relating to the common program which information is to be handed over from said one processor element to said another processor element," as recited in claim 1.

Furthermore, as described in Raz at column 13, lines 26-35,

> From information obtained by said monitoring step, generating workload statistics indicative of the distribution of workload among said plurality of external processors;
> Detecting a workload imbalance in said workload statistics; and
> In response to detecting said workload imbalance, remapping said plurality of volumes to said plurality of connections.

Thus, Raz is describing remapping volumes of a data storage system, not "storing handover information relating to the common program which information is to be handed over from said one processor element to said another processor element," as recited in claim 1.

The seventh clause of claim 1 recites:

> Stopping the performance of said one processor element after said store control section stores said handover information into said storing section.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above. Since neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, they cannot show "stopping the performance of said one processor element after said store control section stores said handover information into said storing section," as recited in claim 1.

The eighth clause of claim 1 recites:

> Starting the performance of said another processor element using said handover information stores in said storing section.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above. Since neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, they cannot show "starting the performance of said another processor element using said handover information stores in said storing section," as recited in claim 1. Claim 1 is submitted to be allowable. Withdrawal of the rejection of claim 1 is earnestly solicited.

Claims 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, and 37-43 depend from claim 1 and add additional distinguishing elements. Claims 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, and 37-43 are thus also submitted to be allowable. Withdrawal of the rejection of claims 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, and 37-43 is earnestly solicited.

Claim 44:

The second clause of claim 44 recites:

> Detecting a switching request signal to request switching such plural processor elements one from another.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "detecting a switching request signal to request switching such plural processor elements, one from another."

Furthermore, persons of ordinary skill in the art would have not been motivated to combine Neches with Raz, as proposed in the final Office Action, as also discussed above with respect to claim 1.

The third clause of claim 44 recites:

> Storing handover information relating to the common program, which information is to be handed over from said one processor element to said another processor element, into a storing section of said multiprocessor system.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "storing handover information relating to the common program, which information is to be handed over from said one

processor element to said another processor element, into a storing section of said multiprocessor system."

The fourth clause of claim 44 recites:

> After said handover information has been stored into the storing section, stopping the performance of said one processor element.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "after said handover information has been stored into the storing section, stopping the performance of said one processor element"

The fifth clause of claim 44 recites:

> Starting the performance of said another processor element using said handover information stored in the storing section.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "starting the performance of said another processor element using said handover information stored in the storing section." Claim 44 ought thus to be allowable as well, for at least those reasons discussed above with respect to claim 1. Withdrawal of the rejection of claim 44 is earnestly solicited.

Claim 45:

The second clause of claim 45 recites:

> Detecting a switching request signal to request switching such plural processor elements one from another.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "detecting a switching request signal to request switching such plural processor elements, one from another."

Furthermore, persons of ordinary skill in the art would have not been motivated to combine Neches with Raz, as proposed in the final Office Action, as also discussed above with respect to claim 1.

The third clause of claim 45 recites:

> Storing handover information relating to the common program, which information is to be handed over from said one processor element to said another processor element, into a storing section of said multiprocessor system.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "storing handover information relating to the common program, which information is to be handed over from said one processor element to said another processor element, into a storing section of said multiprocessor system."

The fourth clause of claim 45 recites:

> After said handover information has been stored into the storing section, stopping the performance of said one processor element.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "after said handover information has been stored into the storing section, stopping the performance of said one processor element."

The fifth clause of claim 45 recites:

> Stopping the performance of said another processor element using said handover information stored in said storing section.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "stopping the performance of said another processor element using said handover information stored in said storing section." Claim 45 ought thus to be allowable as well, for at least those reasons discussed above with respect to claim 1. Withdrawal of the rejection of claim 45 is earnestly solicited.

Claim 46:

The second clause of claim 46 recites:

> A control section for switching between a first processor element and a second processor element during execution of a common program.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "a control section for switching between a first processor element and a second processor element during execution of a common program." Furthermore, persons of ordinary skill in the art would have not been motivated to combine Neches with Raz, as proposed in the final Office Action, as also discussed above with respect to claim 1.

The third clause of claim 46 recites:

A storing section, responsive to each switching of said processor elements by said control section, for storing handover information relating to the execution of the common program by said first processor element before said handover for use by said second processor after said handover.

Neither Neches nor Raz teach, disclose, or suggest switching plural processor elements one from another, as discussed above with respect to claim 1, let alone "a storing section, responsive to each switching of said processor elements by said control section, for storing handover information relating to the execution of the common program by said first processor element before said handover for use by said second processor after said handover." Claim 46 is thus believed to be allowable as well, for at least those reasons discussed above with respect to claim 1. Withdrawal of the rejection of claim 46 is earnestly solicited.

Conclusion:

Accordingly, in view of the reasons given above, it is submitted that all of claims 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, and 37-46 are allowable over the cited references. There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

If there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 25APR06

By: _____
Thomas E. McKiernan
Registration No. 37,889

1201 New York Ave, N.W., 7th Floor
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501